

Progress update

09.03.2021

K. Furman

Investigating the PUB socket binding issue

- This information was in the old ZMQ guide, cut from the current one
- The process of binding a socket is **asynchronous**
- When `socket.bind()` is called a thread for the socket and the correlated queue is started and the binding happens in the background
- Code after the bind call is executed immediately, while it takes up to 1 second to fully set up the socket!
- There are a few ways of dealing with this, but they are all done on the publisher side (i.e. readout card)

Option 1: just wait

- Waiting one (1) second before sending data is a valid solution as that is the maximum time a bind takes
- Even if you do not make a wait command in your code, the PUB socket just drops any data sent before the socket is bound – does not break unless you used a “NO WAIT” option in the socket to raise an error
- Once the socket is bound PUB-SUB is great for continuous readout, and you do not need to rebind the socket during operation – so the problem is only there at the start

Option 2: handshake – “elegant solution”

- Setting up a separate set of REQ-REP sockets to exchange a handshake
- The bind thread will finish in the time a handshake is fully executed
- No significant data must be sent, just an empty message back and forth
- PACMAN already has a REQ server one can use for this approach, not all solutions/setups will

```
cmd_socket.bind(_cmd_server)
data_socket.bind(_data_server)
echo_socket.bind(_echo_server)

# Synchronisation with readout
# Set up a poller, wait for signal from readout to start sending data
poller = zmq.Poller()
poller.register(cmd_socket, zmq.POLLIN)
items = dict(poller.poll())
print("Waiting for signal from readout to start sending data...")
if cmd_socket in items:
    message = cmd_socket.recv()
    print("Signal received.")
    cmd_socket.send(b'')
```

PUB SIDE

```
commander = zmq.Context().socket(zmq.REQ)
commander.connect(cmd)
```

SUB SIDE

```
# Using SUB socket to collect data
reader = zmq.Context().socket(zmq.SUB)
reader.connect(data)
reader.setsockopt(zmq.SUBSCRIBE, b'')

print("Press ENTER to start listening...")
input()
commander.send(b'')
print("Signal sent to PACMAN card.")
commander.recv()
```

Option 3: monitoring socket state

- Setting up a PAIR socket to listen in on the state of your PUB socket (before the bind call)
- Sending data once ZMQ_EVENT_LISTENING returns True
- Seems complicated to set up for what it does, but viable
- Note: Requires using inproc sockets not supported in ZMQ on some operating systems, notably Windows



zmq_socket_monitor(3)

zmq_socket_monitor(3)

[Back to Contents](#)

ØMQ Manual - ØMQ/4.1.7

[+ Table of Contents](#)

Name

zmq_socket_monitor - monitor socket events

Synopsis

```
int zmq_socket_monitor (void *socket, char *endpoint, int events);
```

Description

The `zmq_socket_monitor()` method lets an application thread track socket events (like connects) on a ZeroMQ socket. Each call to this method creates a `ZMQ_PAIR` socket and binds that to the specified `inproc:// endpoint`. To collect the socket events, you must create your own `ZMQ_PAIR` socket, and connect that to the endpoint.

The `events` argument is a bitmask of the socket events you wish to monitor, see *Supported events* below. To monitor all events, use the event value `ZMQ_EVENT_ALL`.

Each event is sent as two frames. The first frame contains an event number (16 bits), and an event value (32 bits) that provides additional data according to the event number. The second frame contains a string that specifies the affected TCP or IPC endpoint.

Option 4: mutex (probably not viable)

- While ZMQ has a `zmutex()` functionality, it is deprecated, not officially supported, and can cause issues
- I do not have experience with mutexes so it may just be my ignorance, but it does not seem viable with how little documentation there is on the ZMQ implementation

Next steps

- Talk with Roland about integration with the DAQ framework (tomorrow!)
- Investigating STREAM type sockets for sending raw bytes to the rest of the DAQ (software bridge) - <http://api.zeromq.org/4-0:zmq-socket#toc19>
- I am out of commission for the next 2 weeks due to STFC HEP Summer School (yes, in March)

The End

Questions?